

PREV PACKAGE    NEXT PACKAGE

**FRAMES      NO FRAMES**

## Interface Summary

## Directory Manager

DirectoryManager provides an interface for accessing Directory data.

### DirectorySource

**DirectorySource** is an interface which provides access to Directory data sources.

### DefaultDirectorySource

### DirectorySource implementation.

### Directory Entry

Represents an LDAP Directory Entry and a LDAP invocation handler used in Proxy instances.

## DirectoryManagerFactory

Use to create an object which implements the `DirectoryManager` interface.

## Generator

Generates java interfaces which represents LDAP object classes.

[illegible]

PREV PACKAGE    NEXT PACKAGE

**FRAMES**      **NO FRAMES**

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

## Package com.gepower.sfo.tool.ldap

### Interface Summary

|   |  |
|---|--|
| <a href="#"><u>DirectoryManager</u></a> | DirectoryManager provides an interface for accessing Directory data.             |
| <a href="#"><u>DirectorySource</u></a>  | DirectorySource is an interface which provides access to Directory data sources. |

### Class Summary

|  |   |
|--|---|
| <a href="#"><u>DefaultDirectorySource</u></a>  | DirectorySource implementation.   |
| <a href="#"><u>DirectoryEntry</u></a>          | Represents an LDAP Directory Entry and a LDAP invocation handler used in Proxy instances. |
| <a href="#"><u>DirectoryManagerFactory</u></a> | Use to create an object which implements the DirectoryManager interface.                  |
| <a href="#"><u>Generator</u></a>               | Generates java interfaces which represents LDAP object classes.                           |

[Overview](#) [Package](#) [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV PACKAGE](#) [NEXT PACKAGE](#)

[FRAMES](#) [NO FRAMES](#)

## Overview Package Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

[SUMMARY](#) [INNER](#) [FIELD](#) [CONSTR](#) [METHOD](#)

[DETAIL](#) [FIELD](#) [CONSTR](#) [METHOD](#)

com.gepower.sfo.tool.ldap

# Interface DirectoryManager

public interface **DirectoryManager**

DirectoryManager provides an interface for accessing Directory data.

## Method Summary

|                  |  |
|------------------|--|
| java.lang.Object | <b>cast</b> (java.lang.Object entry, java.lang.Class interfaceToCastTo)<br>This method provides backward compatibility from Java 1.3 to Java 1.2 which does not support the Proxy class.   |
| java.lang.String | <b>getDN</b> (java.lang.Object entry)<br>Use to get the specified 'entry' distinguished name.  |
| java.lang.Object | <b>lookup</b> (java.lang.String dn)<br>Use to lookup a specific entry identified by the specified 'dn'.  |
| java.lang.Object | <b>mixinInterfaces</b> (java.lang.Object entry, java.lang.Class[] newInterfaces)<br>Use to add additional interfaces specified by 'newInterfaces' to an existing 'entry'.  |
| java.lang.Object | <b>newEntryInstance</b> (java.lang.String dn, java.lang.Class[] interfaces)<br>Use to create a new LDAP entry.   |
|                  | <b>remove</b> (java.lang.Object entry)<br>Use to remove an existing entry from the Directory.  |
| java.util.List   | <b>search</b> (java.lang.String ctxToSearch, java.lang.String filter)<br>Use to execute a query against the Directory using the specified 'ctxToSearch', and 'filter'.   |
| java.util.List   | <b>search</b> (java.lang.String ctxToSearch, java.lang.String filter, javax.naming.directory.SearchControls searchCtrls)<br>Use to execute a query against the Directory using the specified 'ctxToSearch', 'filter', and 'searchCtrls'.   |
| java.util.List   | <b>search</b> (java.lang.String ctxToSearch, java.lang.String filter, javax.naming.directory.SearchControls searchCtrls, javax.naming.ldap.Control[] reqCtrls)<br>Use to execute a query against the Directory using the specified 'ctxToSearch', 'filter', 'searchCtrls', and 'reqCtrls'. |

**write**(java.lang.Object entry)

Use to commit a new entry or modifications of an existing entry to the Directory.

## Method Detail

### newEntryInstance

```
public java.lang.Object newEntryInstance(java.lang.String dn,
                                         java.lang.Class[] interfaces)
                                         throws javax.naming.NamingException
```

Use to create a new LDAP entry. The entry is not written to the Directory until DirectoryManager.write() is executed.

#### Parameters:

dn - Distinguished name for the new entry. Must not be null or empty.

interfaces - Array of Class objects which represent the interfaces that this new entry will support. The Class objects MUST be one of the LDAP code generated interfaces. Array must not be null or empty.

#### Returns:

Object representing the directory entry. This object can be cast to the appropriate "objectclass" interface (s).

#### Throws:

javax.naming.NamingException - if a naming exception is encountered.

### mixinInterfaces

```
public java.lang.Object mixinInterfaces(java.lang.Object entry,
                                         java.lang.Class[] newInterfaces)
                                         throws javax.naming.NamingException
```

Use to add additional interfaces specified by 'newInterfaces' to an existing 'entry'. The modified entry is not written to the Directory until DirectoryManager.write() is executed.

#### Parameters:

entry - Existing LDAP entry to mix new interfaces into. 'entry' must be acquired by calls to DirectoryController.lookup(), DirectoryController.search(), or DirectoryController.newEntryInstance().

newInterfaces - Array of new interfaces to mix into the entry. Must not be null and must not be empty.

#### Returns:

Object representing the modified directory entry. This object can be cast to the appropriate "objectclass" interface(s) including those contained in 'newInterfaces'.

#### Throws:

javax.naming.NamingException - if a naming exception is encountered.

### lookup

```
public java.lang.Object lookup(java.lang.String dn)
    throws javax.naming.NameNotFoundException,
           javax.naming.NamingException,
           java.lang.ClassNotFoundException
```

Use to lookup a specific entry identified by the specified 'dn'.

**Parameters:**

dn - The distinguished name which uniquely identifies the entry. Must not be null and must not be empty.

**Returns:**

Object representing the directory entry bound to the specified dn. This object can be cast to the appropriate "objectclass" interface(s).

**Throws:**

javax.naming.NameNotFoundException - if dn cannot be resolved because it is not bound  
 javax.naming.NamingException - if a naming exception is encountered.  
 java.lang.ClassNotFoundException - if the looked up entry contains an object class which does not have an associated code generated interface.

## search

```
public java.util.List search(java.lang.String ctxToSearch,
    java.lang.String filter)
    throws javax.naming.NamingException,
           java.lang.ClassNotFoundException
```

Use to execute a query against the Directory using the specified 'ctxToSearch', and 'filter'.

**Parameters:**

ctxToSearch - Context to search. "" for current context. Must not be null.  
 filter - LDAP search filter. Must not be null.

**Returns:**

List of Objects representing the results of the search. These Object can each be cast to the appropriate "objectclass" interface(s). If search finds nothing, List returned will have size of zero. Return will never be null.

**Throws:**

java.lang.ClassNotFoundException - if the entries found contains an object class which does not have an associated code generated interface.  
 javax.naming.NamingException - if naming exception is encountered.

## search

```
public java.util.List search(java.lang.String ctxToSearch,
    java.lang.String filter,
    javax.naming.directory.SearchControls searchCtrls)
    throws javax.naming.NamingException,
           java.lang.ClassNotFoundException
```

Use to execute a query against the Directory using the specified 'ctxToSearch', 'filter', and 'searchCtrls'.

**Parameters:**

ctxToSearch - Context to search. "" for current context. Must not be null.  
 filter - LDAP search filter. Must not be null.  
 searchCtrls - Used to determine scope of search and what gets returned. May be null. If null, defaults will be used (search using SearchControls.SUBTREE\_SCOPE).

**Returns:**

List of Objects representing the results of the search. These Object can each be cast to the appropriate "objectclass" interface(s). If search finds nothing, List returned will have size of zero. Return will never be null.

**Throws:**

java.lang.ClassNotFoundException - if the entries found contains an object class which does not have an associated code generated interface.  
 javax.naming.NamingException - if naming exception is encountered.

**See Also:**

SearchControls

## search

```
public java.util.List search(java.lang.String ctxToSearch,
                             java.lang.String filter,
                             javax.naming.directory.SearchControls searchCtrls,
                             javax.naming.ldap.Control[] reqCtrls)
    throws javax.naming.NamingException,
           java.lang.ClassNotFoundException
```

Use to execute a query against the Directory using the specified 'ctxToSearch', 'filter', 'searchCtrls', and 'reqCtrls'.

**Parameters:**

ctxToSearch - Context to search. "" for current context. Must not be null.  
 filter - LDAP search filter. Must not be null.  
 searchCtrls - Used to determine scope of search and what gets returned. May be null. If null, defaults will be used (search using SearchControls.SUBTREE\_SCOPE).  
 reqCtrls - A control to request the LDAP search to return in a certain way (i.e, sort results in a particular way). May be null. If null, no LDAP request controls will be used.

**Returns:**

List of Objects representing the results of the search. These Object can each be cast to the appropriate "objectclass" interface(s). If search finds nothing, List returned will have size of zero. Return will never be null.

**Throws:**

java.lang.ClassNotFoundException - if the entries found contains an object class which does not have an associated code generated interface.  
 javax.naming.NamingException - if naming exception is encountered.

**See Also:**

SearchControls, Control

## write

```
public void write(java.lang.Object entry)
    throws javax.naming.NamingException
```

Use to commit a new entry or modifications of an existing entry to the Directory.

### Parameters:

entry - Entry to commit to the directory. 'entry' must have been acquired by calls to DirectoryController.lookup(), DirectoryController.search(), or DirectoryController.newEntryInstance(). Must not be null.

### Throws:

javax.naming.NamingException - if naming exception is encountered.

## remove

```
public void remove(java.lang.Object entry)
    throws javax.naming.NamingException
```

Use to remove an existing entry from the Directory.

### Parameters:

entry - Entry to remove from the directory. 'entry' must have been acquired by calls to DirectoryController.lookup(), DirectoryController.search(), or DirectoryController.newEntryInstance(). Must not be null.

### Throws:

javax.naming.NamingException - if naming exception is encountered.

## cast

```
public java.lang.Object cast(java.lang.Object entry,
    java.lang.Class interfaceToCastTo)
    throws java.lang.ClassCastException
```

This method provides backward compatibility from Java 1.3 to Java 1.2 which does not support the Proxy class. This method is not yet implemented.

### Parameters:

entry - Entry to cast. 'entry' must have been acquired by calls to DirectoryController.lookup(), DirectoryController.search(), or DirectoryController.newEntryInstance(). Must not be null.  
interfaceToCastTo - This is the interface that the specified 'entry' is to be cast to.

### Returns:

Object which can be cast to the type specified by 'interfaceToCastTo'.

### Throws:

java.lang.ClassCastException - if the specified 'entry' cannot be cast to the specified

'interfaceToCastTo'.

---

## getDN

```
public java.lang.String getDN(java.lang.Object entry)
```

Use to get the specified 'entry' distinguished name.

**Parameters:**

entry - Entry to obtain distinguished name from. 'entry' must have been acquired by calls to DirectoryController.lookup(), DirectoryController.search(), or DirectoryController.newEntryInstance(). Must not be null.

**Returns:**

String containing the specified 'entry' distinguished name.

---

## [Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



**Overview Package Class Tree Deprecated Index Help**

PREV CLASS NEXT CLASS

FRAMES NO FRAMES

SUMMARY: INNER | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

com.gepower.sfo.tool.ldap

**Class DefaultDirectorySource**

java.lang.Object

---com.gepower.sfo.tool.ldap.DefaultDirectorySource

**All Implemented Interfaces:**DirectorySourcepublic class **DefaultDirectorySource**

extends java.lang.Object

implements DirectorySource

DirectorySource implementation. See DirectorySource for discription of implemented methods.

**Constructor Summary**DefaultDirectorySource (java.util.Hashtable environment)**Method Summary**

|                                   |   |
|-----------------------------------|---|
| void                              | <u>discardDirContext</u> (javax.naming.directory.DirContext context)<br>Use to discard the specified 'context'. |
| javax.naming.directory.DirContext | <u>getDirContext</u> ()<br>Use to get a JNDI DirContext object.   |
| void                              | <u>releaseDirContext</u> (javax.naming.directory.DirContext context)<br>Use to release the specified 'context'. |

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

## DefaultDirectorySource

```
public DefaultDirectorySource(java.util.Hashtable environment,  
                             throws javax.naming.NamingException
```

### Method Detail

#### getDirContext

```
public javax.naming.directory.DirContext getDirContext()  
                                         throws javax.naming.NamingException
```

**Description copied from interface: DirectorySource**

Use to get a JNDI DirContext object.

**Specified by:**

getDirContext in interface DirectorySource

Following copied from interface: com.gepower.sfo.tool.ldap.DirectorySource

**Returns:**

DirContext object.

**Throws:**

javax.naming.NamingException - if a naming exception is encountered.

---

#### releaseDirContext

```
public void releaseDirContext(javax.naming.directory.DirContext context)
```

**Description copied from interface: DirectorySource**

Use to release the specified 'context'. This should be called when the context is no longer needed.

**Specified by:**

releaseDirContext in interface DirectorySource

Following copied from interface: com.gepower.sfo.tool.ldap.DirectorySource

**Parameters:**

context - The context to release.

---

#### discardDirContext

```
public void discardDirContext(javax.naming.directory.DirContext context)
```

**Description copied from interface: DirectorySource**

Use to discard the specified 'context'.

**Specified by:**

discardDirContext in interface DirectorySource

**context** - The context to release.

PREV CLASS NEXT CLASS

SUMMARY. INNER | FIELD | CONSTR | METHOD

FRAMES      NO FRAMES

DETAIL: FIELD | CONSTR : METHOD

**[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)**[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#)[SUMMARY](#) [INNER](#) [FIELD](#) [CONSTR](#) [METHOD](#)[DETAIL](#) [FIELD](#) [CONSTR](#) [METHOD](#)

com.gepower.sfo.tool.ldap

**Class DirectoryEntry**

java.lang.Object

+---com.gepower.sfo.tool.ldap.DirectoryEntry

**All Implemented Interfaces:**

java.lang.reflect.InvocationHandler, java.io.Serializable

public class **DirectoryEntry**

extends java.lang.Object

implements java.io.Serializable, java.lang.reflect.InvocationHandler

Represents an LDAP Directory Entry and a LDAP invocation handler used in Proxy instances. Each proxy instance has an associated invocation handler. When a method is invoked on a proxy instance, the method invocation is encoded and dispatched to the invoke method of its invocation handler. This is a package scope class and not used directly by clients.

**See Also:**InvocationHandler, java.lang.reflect.Proxy, [Serialized Form](#)**Method Summary**

|                  |  |
|------------------|--|
| java.lang.Object | <b>invoke</b> (java.lang.Object proxy, java.lang.reflect.Method method, java.lang.Object[] args)<br>Implement abstract method invoke() from InvocationHandler. |
| java.lang.String | <b>toString</b> (java.lang.Object proxy, java.lang.reflect.Method method)<br>Returns the contents of all attribute in this entry.                              |

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Method Detail**

## invoke

```
public java.lang.Object invoke(java.lang.Object proxy,
                                java.lang.reflect.Method method,
                                java.lang.Object[] args)
    throws java.lang.Throwable
```

Implement abstract method `invoke()` from `InvocationHandler`. This method only recognizes methods that have been declared in the generated interfaces.

### Specified by:

`invoke` in interface `java.lang.reflect.InvocationHandler`

### Parameters:

`proxy` - the proxy instance that the method was invoked on.

`method` - the `Method` instance corresponding to the interface method invoked on the proxy instance. The declaring class of the `Method` object will be the interface that the method was declared in, which may be a superinterface of the proxy interface that the proxy class inherits the method through.

`args` - an array of objects containing the values of the arguments passed in the method invocation on the proxy instance, or null if interface method takes no arguments. Arguments of primitive types are wrapped in instances of the appropriate primitive wrapper class, such as `java.lang.Integer` or `java.lang.Boolean`.

### Throws:

`java.lang.Throwable` - the exception to throw from the method invocation on the proxy instance. The exception's type must be assignable either to any of the exception types declared in the throws clause of the interface method or to the unchecked exception types `java.lang.RuntimeException` or `java.lang.Error`. If a checked exception is thrown by this method that is not assignable to any of the exception types declared in the throws clause of the interface method, then an `UndeclaredThrowableException` containing the exception that was thrown by this method will be thrown by the method invocation or the proxy instance.

### See Also:

`java.lang.reflect.UndeclaredThrowableException`

## toString

```
public java.lang.String toString(java.lang.Object proxy,
                                   java.lang.reflect.Method method)
```

Returns the contents of all attribute in this entry. Use for debugging purposes only.

### Parameters:

`proxy` - The Proxy object serviced by this `InvocationHandler`.

`method` - The `Method` object invoked on the Proxy.

### Returns:

The contents of all attributes in this entry.

**Overview Package Class Tree Deprecated Index Help**

**PREV CLASS NEXT CLASS**

**FRAMES NO FRAMES**

**SUMMARY: INNER | FIELD | CONSTR | METHOD**

**DETAIL: FIELD | CONSTR | METHOD**

14  
Page 3 of 3

**[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)**[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#)[SUMMARY](#) [INNER](#) [FIELD](#) [CONSTR](#) [METHOD](#)[DETAIL](#) [FIELD](#) [CONSTR](#) [METHOD](#)

com.gepower.sfo.tool.ldap

**Class DirectoryManagerFactory**

java.lang.Object

+---com.gepower.sfo.tool.ldap.DirectoryManagerFactory

```

public class DirectoryManagerFactory
    extends java.lang.Object

```

Use to create an object which implements the DirectoryManager interface.

**Constructor Summary****[DirectoryManagerFactory\(\)](#)****Method Summary**

|  |   |
|--|---|
| <b><a href="#">newDirectoryManager()</a></b>                             | <b><a href="#">newDirectoryManager</a></b> ( <a href="#">DirectorySource</a> src, java.lang.String pkg)<br>Creates a new object which implements the DirectoryManager interface using the specified 'src' and 'pkg'.  |
| <b><a href="#">newDirectoryManager(java.lang.ClassLoader loader)</a></b> | <b><a href="#">newDirectoryManager</a></b> ( <a href="#">DirectorySource</a> src, java.lang.String pkg, java.lang.ClassLoader loader)<br>Creates a new object which implements the DirectoryManager interface using the specified 'src', 'pkg', and 'loader'.                                       |
| <b><a href="#">newDirectoryManager(java.io.PrintStream logger)</a></b>   | <b><a href="#">newDirectoryManager</a></b> ( <a href="#">DirectorySource</a> src, java.lang.String pkg, java.lang.ClassLoader loader, java.io.PrintStream logger)<br>Creates a new object which implements the DirectoryManager interface using the specified 'src', 'pkg', 'loader', and 'logger'. |
| <b><a href="#">newDirectoryManager(java.util.Hashtable env)</a></b>      | <b><a href="#">newDirectoryManager</a></b> (java.util.Hashtable env, java.lang.String pkg)<br>Creates a new object which implements the DirectoryManager interface using the specified 'env' and 'pkg'.   |

|   |   |
|---|---|
| <code>static <u>DirectoryManager</u></code> | <b><code>newDirectoryManager</code></b> (java.util.Hashtable env, java.lang.String pkg, java.lang.ClassLoader loader)<br>Creates a new object which implements the DirectoryManager interface using the specified 'env', 'pkg', and 'loader'.                                       |
| <code>static <u>DirectoryManager</u></code> | <b><code>newDirectoryManager</code></b> (java.util.Hashtable env, java.lang.String pkg, java.lang.ClassLoader loader, java.io.PrintStream logger)<br>Creates a new object which implements the DirectoryManager interface using the specified 'env', 'pkg', 'loader', and 'logger'. |

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### DirectoryManagerFactory

```
public DirectoryManagerFactory()
```

## Method Detail

### newDirectoryManager

```
public static DirectoryManager newDirectoryManager(java.util.Hashtable env,
                                                    java.lang.String pkg)
                                                    throws java.lang.IllegalArgumentException,
                                                    javax.naming.NamingException
```

Creates a new object which implements the DirectoryManager interface using the specified 'env' and 'pkg'.

#### Parameters:

env - Used to specify various preferences and properties that define the environment in which naming and directory services are accessed. Must not be null.

pkg - The java package in which the LDAP interfaces were generated under. Must not be null.

#### Throws:

java.lang.IllegalArgumentException - if 'env' or 'pkg' is null.

javax.naming.NamingException - if a naming exception is encountered.

### newDirectoryManager

```
public static DirectoryManager newDirectoryManager(java.util.Hashtable env,
                                                    java.lang.String pkg,
                                                    java.lang.ClassLoader loader)
                                                    throws java.lang.IllegalArgumentException,
```



javax.naming.NamingException

Creates a new object which implements the DirectoryManager interface using the specified 'env', 'pkg', and 'loader'.

**Parameters:**

env - Used to specify various preferences and properties that define the environment in which naming and directory services are accessed. Must not be null.  
 pkg - The java package in which the LDAP interfaces were generated under. Must not be null.  
 loader - Class loader to use to load proxy classes. May be null in which case the current threads class loader will be used.

**Throws:**

java.lang.IllegalArgumentException - if 'env' or 'pkg' is null.  
 javax.naming.NamingException - if a naming exception is encountered.

## newDirectoryManager

```
public static DirectoryManager newDirectoryManager(java.util.Hashtable env,
                                                    java.lang.String pkg,
                                                    java.lang.ClassLoader loader,
                                                    java.io.PrintStream logger)
    throws java.lang.IllegalArgumentException,
           javax.naming.NamingException
```

Creates a new object which implements the DirectoryManager interface using the specified 'env', 'pkg', 'loader', and 'logger'.

**Parameters:**

env - Used to specify various preferences and properties that define the environment in which naming and directory services are accessed. Must not be null.  
 pkg - The java package in which the LDAP interfaces were generated under. Must not be null.  
 loader - Class loader to use to load proxy classes. May be null in which case the current threads class loader will be used.  
 logger - This is where all debug trace messages will be written to.

**Throws:**

java.lang.IllegalArgumentException - if 'env' or 'pkg' is null.  
 javax.naming.NamingException - if a naming exception is encountered.

## newDirectoryManager

```
public static DirectoryManager newDirectoryManager(DirectorySource src,
                                                    java.lang.String pkg)
    throws java.lang.IllegalArgumentException,
           javax.naming.NamingException
```

Creates a new object which implements the DirectoryManager interface using the specified 'src' and 'pkg'.

**Parameters:**

src - Specifies the what directory source the DirectoryManager will use. Must not be null.

pkg - The java package in which the LDAP interfaces were generated under. Must not be null.

**Throws:**

java.lang.IllegalArgumentException - if 'src' or 'pkg' is null.  
 javax.naming.NamingException - if a naming exception is encountered.

## newDirectoryManager

```
public static DirectoryManager newDirectoryManager(DirectorySource src,
                                                    java.lang.String pkg,
                                                    java.lang.ClassLoader loader,
                                                    throws java.lang.IllegalArgumentException,
                                                    javax.naming.NamingException)
```

Creates a new object which implements the DirectoryManager interface using the specified 'src', 'pkg', and 'loader'.

**Parameters:**

src - Specifies the what directory source the DirectoryManager will use. Must not be null.  
 pkg - The java package in which the LDAP interfaces were generated under. Must not be null.  
 loader - Class loader to use to load proxy classes. May be null in which case the current threads class loader will be used.

**Throws:**

java.lang.IllegalArgumentException - if 'src' or 'pkg' is null.  
 javax.naming.NamingException - if a naming exception is encountered.

## newDirectoryManager

```
public static DirectoryManager newDirectoryManager(DirectorySource src,
                                                    java.lang.String pkg,
                                                    java.lang.ClassLoader loader,
                                                    java.io.PrintStream logger)
                                                    throws java.lang.IllegalArgumentException,
                                                    javax.naming.NamingException
```

Creates a new object which implements the DirectoryManager interface using the specified 'src', 'pkg', 'loader', and 'logger'.

**Parameters:**

src - Specifies the what directory source the DirectoryManager will use. Must not be null.  
 pkg - The java package in which the LDAP interfaces were generated under. Must not be null.  
 loader - Class loader to use to load proxy classes. May be null in which case the current threads class loader will be used.  
 logger - This is where all debug trace messages will be written to.

**Throws:**

java.lang.IllegalArgumentException - if 'src' or 'pkg' is null.  
 javax.naming.NamingException - if a naming exception is encountered.



**PREV CLASS**    **NEXT CLASS**

**FRAMES**      **NO FRAMES**

DETAIL: FIELD | CONSTR | METHOD

[Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.gepower.sfo.tool.ldap

## Class Generator

java.lang.Object

---com.gepower.sfo.tool.ldap.Generator

public abstract class **Generator**  
extends java.lang.Object

Generates java interfaces which represents LDAP object classes. These classes are used in the java LDAP Directory framework. This class is abstract can contains only static methods. This class contains a main() method and is designed to be executed from the command line. See method description for main() for more details.

### See Also:

[main\(java.lang.String\[\]\)](#)

## Constructor Summary

[Generator\(\)](#)

## Method Summary

|             |  |
|-------------|--|
| static void | <a href="#">main</a> (java.lang.String[] args)<br>Usage: java com.gepower.sfo.tool.ldap.Generator params [options] |
|-------------|--|

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Generator

public **Generator**()

## Method Detail

### main

```
public static void main(java.lang.String[] args)
```

Usage: java com.gepower.sfo.tool.ldap.Generator params [options]

To print out help, use the -help option when executing this program from the command line.

#### Parameters:

args - Array of String arguments which consists of the required and optional parameters.

#### Required Parameters:

- '-sourcerootpath' the root directory path for the generated java source.
- '-package' the java package for the generated java source.
- '-directxfactory' class to use for the initial directory context factory.
- '-providerurl' the LDAP URL string (i.e., ldap://localhost:389/o=ge.com).
- '-securityprincipal' identity of the principal for authenticating the caller to the service.
- '-securitycredentials' credentials of the principal for authenticating the caller to the service.
- '-securityauthentication' security level to use.

#### Optional Parameters:

- '-exclude' object classes matching the wildcard will be excluded from code generation. Exclusions have precedence over Inclusion. Multiple wildcards can be specified separated by semi-colons. (i.e. "ns\*; ob\*; net\*server").
- '-include' object classes matching the wildcard will be included in code generation. If option not specified, include all object classes. Multiple wildcards can be specified. see exclude option.
- '-version' version number that will be included into the javadoc of the generated code.
- '-tabstop' tab stop to use when formatting the generated code.
- '-help' use to print usage syntax on the command line.
- '-?' use to print usage syntax on the command line.

---

## [Overview](#) [Package](#) [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---